

```

EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTTTTTTTTTTTTTTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEEEEEEEEEEEEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEE DDD DDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT
EEEEEEEEEEEEEEEEEE DDDDDDDDDDDDDD TTT

```

**EXE**

Mod

ED 1

ED

ED

ED

ED

ED1

ED

ED

ED  
EDED  
EDSYS  
LIB

L18

```

FFFFFFFFF  IIIII  LL      EEEEEEEEE  IIIII  000000
FFFFFFFFF  IIIII  LL      EEEEEEEEE  IIIII  000000
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FFFFFFF    II    LL      EEEEEEEE   II    00      00
FFFFFFF    II    LL      EEEEEEEE   II    00      00
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FF         II    LL      EE          II    00      00
FF         IIIII  LLLLLLLLL  EEEEEEEEE  IIIII  000000
FF         IIIII  LLLLLLLLL  EEEEEEEEE  IIIII  000000
                                     ....
                                     ....
                                     ....
                                     ....

LL         IIIII  SSSSSSSS
LL         IIIII  SSSSSSSS
LL         II    SS
LL         II    SS
LL         II    SS
LL         II    SS
LL         II    SSSSSS
LL         II    SSSSSS
LL         II    SS
LL         II    SS
LL         II    SS
LL         IIIII  SSSSSSSS
LLLLLLLLLL IIIII  SSSSSSSS
LLLLLLLLLL IIIII  SSSSSSSS

```



```
0001 0 %TITLE 'FILEIO - Central file I/O module'
0002 0 MODULE EDT$FILEIO (
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1
0031 1 ++
0032 1 FACILITY: EDT -- The DEC Standard Editor
0033 1
0034 1 ABSTRACT:
0035 1
0036 1 This is the central file i/o routine used by EDT.
0037 1
0038 1 ENVIRONMENT: Runs in user mode on VAX/VMS and non-privileged PDP-11
0039 1
0040 1 AUTHOR: Shelly T. Solomon, CREATION DATE: 07-Dec-1981
0041 1
0042 1 MODIFIED BY:
0043 1
0044 1 1-001 - Original. STS 25-Dec-1981
0045 1 1-002 - Change module name to EDT$FILEIO. STS 25-Dec-1981
0046 1 1-003 - Add calls for include file. STS 26-Dec-1981
0047 1 1-004 - Add require files for 11 translations. STS 28-Dec-1981
0048 1 1-005 - Add linkage attribute to routine. STS 30-Dec-1981
0049 1 1-006 - Signal any errors. STS 06-Jan-1982
0050 1 1-007 - Add code for opening output file. STS 13-Jan-1982
0051 1 1-008 - Fix DSC$A_POINTER macro STS 14-Jan-1982
0052 1 1-009 - Add gets and puts STS 15-Jan-1982
0053 1 1-010 - Change opening journal file to open in-out. STS 18-Jan-1982
0054 1 1-011 - Fixed undefined symbol EDT$$opn_inout on 11. STS 19-Jan-1982
0055 1 1-012 - output filenames with error messages. STS 19-Jan-1982
0056 1 1-013 - Change the defaulting of the journal file name. STS 21-Jan-1982
0057 1 1-014 - Add check to see if file is VFC format. STS 22-Jan-1982
```

58 0058 1 1-015 - fix journal file name for 11's. STS 26-Jan-1982  
59 0059 1 1-016 - Add dot to sequence parameter passed with journal file.  
60 0060 1 STS 28-Jan-1982  
61 0061 1 1-017 - Pass RHB info down to 11 i/o routines. STS 02-Feb-1982  
62 0062 1 1-018 - Take out extra dot in get on 11's, also  
63 0063 1 fix include rab. STS 08-Feb-1982  
64 0064 1 1-019 - add flush for journal buffer. STS 11-Feb-1982  
65 0065 1 1-020 - Take out call to edt\$\$get\_fnam. STS 12-Feb-1982  
66 0066 1 1-021 - Pass correct status back to caller. STS 26-Feb-1982  
67 0067 1 1-022 - Add literals for callable parameters. STS 08-Mar-1982  
68 0068 1 1-023 - Fix status passed on opening write file. STS 10-Mar-1982  
69 0069 1 1-024 - Rearrange interface to EDT\$IOMOD to improve the rationality  
70 0070 1 of file naming. JBS 25-Mar-1982  
71 0071 1 1-025 - Worry about non-standard input files. JBS 26-Mar-1982  
72 0072 1 1-026 - Correct a typo. JBS 27-Mar-1982  
73 0073 1 1-027 - Make the new file handling logic work on the PDP-11. JBS 29-Mar-1982  
74 0074 1 1-028 - Use temporary file for WRITE and EXIT and then Rename it. SMB 31-Mar-1982  
75 0075 1 1-029 - Add related file names for the PDP-11. JBS 31-Mar-1982  
76 0076 1 1-030 - Distinguish two cases of output open for journal files on the PDP-11  
77 0077 1 and add a flush counter to improve PDP-11 performance. JBS 01-Apr-1982  
78 0078 1 1-031 - Rearrange file name handling for the journal file. JBS 02-Apr-1982  
79 0079 1 1-032 - Make more modifications for WRITE/EXIT to temp files. SMB 02-Apr-1982  
80 0080 1 1-033 - Cannot use %REF in STRING\_DESC. JBS 03-Apr-1982  
81 0081 1 1-034 - Fix bugs in PDP-11 opening of output files. SMB 06-Apr-1982  
82 0082 1 1-035 - Add rename for PDP-11's and CLOSE\_DEL for output files. SMB 08-Apr-1982  
83 0083 1 1-036 - Convert PDP-11 filenames to uppercase. SMB 12-Apr-1982  
84 0084 1 1-037 - Take out fix 1-036(move to LWRITE)-fix error message filename for VAX. SMB 13-Apr-1982  
85 0085 1 1-038 - Always return status when closing PDP-11 files. JBS 09-Apr-1982  
86 0086 1 1-039 - Reverse the attributes flag. JBS 12-Apr-1982  
87 0087 1 1-040 - Merge the last four edits, which were done independently. JBS 15-Apr-1982  
88 0088 1 1-041 - Add a parse before opening output files. SMB 15-Apr-1982  
89 0089 1 1-042 - Put back line accidentally deleted for filename storage. SMB 16-Apr-1982  
90 0090 1 1-043 - Conditionalize the conversion to uppercase. SMB 22-Apr-1982  
91 0091 1 1-044 - Restrict renaming to disks or DECtapes only. SMB 26-Apr-1982  
92 0092 1 1-045 - Change the ordinals of global literals for file types. SMB 19-May-1982  
93 0093 1 1-046 - Add some comments. STS 19-May-1982  
94 0094 1 1-047 - Clean up the magic numbers. JBS 25-May-1982  
95 0095 1 1-048 - Don't use special linkage on 11's. STS 03-Jun-1982  
96 0096 1 1-049 - On OPEN, use RHB as the default name. Also, don't use special linkages on  
97 0097 1 VAX either, since the special linkage used by CALLFIO is compatible with  
98 0098 1 the standard VAX/VMS linkage conventions. JBS 15-Jun-1982  
99 0099 1 1-050 - Implement the new file defaulting rules. JBS 17-Jun-1982  
100 0100 1 1-051 - Signal any bad status from flushing the journal file. STS 30-Jun-1982  
101 0101 1 1-052 - Fix bad parameter pass in open for output without related names. SMB 06-Jul-1982  
102 0102 1 1-053 - Add a special check for RSTS disk files. SMB 07-Jul-1982  
103 0103 1 1-054 - Store status on PDP-11 open for output. SMB 19-Jul-1982  
104 0104 1 1-055 - Check for errors when deleting the journal file. JBS 22-Feb-1983  
105 0105 1 1-056 - Don't maximize version number on WRITE. JBS 04-Apr-1983  
106 0106 1 1-057 - Fix a typo in PDP-11 output file opening. JBS 06-Apr-1983  
107 0107 1 1-058 - Fix the message given when the journal file fails to open for output. JBS 02-May-1983  
108 0108 1 1-059 - Improve the appearance of the listing. JBS 14-Jun-1983  
109 0109 1 1-060 - On VMS, if the EXIT file name is empty,  
110 0110 1 use the resultant name from opening the input file. JBS 29-Jul-1983  
111 0111 1 1-061 - Fix bug in edit 060--the input name was being discarded too soon if the  
112 0112 1 output open happened after the input file was closed. JBS 31-Aug-1983  
113 0113 1 1-062 - Complete edit 061 by storing the input file name even if the file  
114 0114 1 does not open. JBS 06-Sep-1983



EDT\$FILEIO  
V04-000

FILEIO - Central file I/O module

L 11  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILEIO.BLI;1 Page 3  
(1)

: 115  
: 116

0115 1 !--  
0116 1

```

118 0117 1 %SBTTL 'Declarations'
119 0118 1
120 0119 1 | TABLE OF CONTENTS:
121 0120 1 |
122 0121 1 |
123 0122 1 REQUIRE 'EDT$SRC:TRAROUNAM';
124 0561 1
125 0562 1 FORWARD ROUTINE
126 0563 1 | EDT$FILEIO;
127 0564 1 |
128 0565 1 |
129 0566 1 | INCLUDE FILES:
130 0567 1 |
131 0568 1 |
132 0569 1 REQUIRE 'EDT$SRC:EDTREQ';
133 0704 1
134 L 0705 1 %IF %BLISS (BLISS32)
135 0706 1 %THEN
136 0707 1
137 0708 1 REQUIRE 'EDT$SRC:SYSSYM';
138 0738 1
139 0739 1 %FI
140 0740 1
141 0741 1 |
142 0742 1 | MACROS:
143 0743 1 |
144 0744 1 | +
145 0745 1 | Macro for the file type used as a constant. This is defined as a macro
146 0746 1 | so we can use %CHARCOUNT to pass the length of the string.
147 0747 1 | -
148 0748 1 | <BLF/NOFORMAT>
149 0749 1 |
150 0750 1 MACRO
151 0751 1 | TEMP_TYP = '.TMP' %; ! File type for temporary output files (before being renamed)
152 0752 1 |
153 0753 1 | <BLF/FORMAT>
154 0754 1 |
155 0755 1 | EQUATED SYMBOLS:
156 0756 1 |
157 0757 1 |
158 L 0758 1 %IF %BLISS (BLISS32)
159 0759 1 %THEN
160 0760 1
161 0761 1 LITERAL
162 0762 1 | EDT$K_FAC_NO = 133;
163 0763 1 |
164 0764 1 %FI
165 0765 1 |
166 0766 1 | +
167 0767 1 | These codes need to be defined here because they need to be known at compile
168 0768 1 | time in order to be used in case statements
169 0769 1 | -
170 0770 1 |
171 0771 1 GLOBAL LITERAL
172 0772 1 | EDT$K_OPEN_INPUT = 1, ! code signifying we wish to open a file for input
173 0773 1 | EDT$K_OPEN_OUTPUT_SEQ = 2, ! code signifying we wish to open a sequenced file for output
174 0774 1 | EDT$K_OPEN_OUTPUT_NOSEQ = 3, ! code meaning we wish to open a non-sequenced file for output
```



```

: 175      0775 1      EDT$K_OPEN_IN_OUT = 4,
: 176      0776 1      EDT$K_GET = 5,
: 177      0777 1      EDT$K_PUT = 6,
: 178      0778 1      EDT$K_CLOSE_DEL = 7,
: 179      0779 1      EDT$K_CLOSE = 8,
: 180      0780 1      EDT$K_COMMAND_FILE = 1,
: 181      0781 1      EDT$K_INPUT_FILE = 2,
: 182      0782 1      EDT$K_INCLUDE_FILE = 3,
: 183      0783 1      EDT$K_JOURNAL_FILE = 4,
: 184      0784 1      EDT$K_OUTPUT_FILE = 5,
: 185      0785 1      EDT$K_WRITE_FILE = 6;
: 186      0786 1
: 187      0787 1 LITERAL
: 188      0788 1      FLUSH_LIMIT = 5;
: 189      0789 1
: 190      0790 1
: 191      0791 1
: 192      0792 1
: 193      0793 1
: 194      0794 1
: 195      0795 1 LITERAL
: 196      0796 1      DISK_FILE_NO = 0,
: 197      0797 1      DISK_FILE_YES = 1,
: 198      0798 1      DISK_FILE_RSTS = 2,
: 199      0799 1      SEQ_NO = 0,
: 200      0800 1      SEQ_YES = 1,
: 201      0801 1      RELAT_NONE = 0,
: 202      0802 1      RELAT_INPUT = 1,
: 203      0803 1      ATTR_INPUT = 0,
: 204      0804 1      ATTR_DEFAULT = 1,
: 205      0805 1      ATTR_JOURNAL = 2;
: 206      0806 1
: 207      0807 1
: 208      0808 1
: 209      0809 1
: 210      0810 1
: 211      0811 1
: 212      0812 1
: 213      0813 1
: 214      0814 1
: 215      0815 1

      EDT$K_OPEN_IN_OUT = 4,
      EDT$K_GET = 5,
      EDT$K_PUT = 6,
      EDT$K_CLOSE_DEL = 7,
      EDT$K_CLOSE = 8,
      EDT$K_COMMAND_FILE = 1,
      EDT$K_INPUT_FILE = 2,
      EDT$K_INCLUDE_FILE = 3,
      EDT$K_JOURNAL_FILE = 4,
      EDT$K_OUTPUT_FILE = 5,
      EDT$K_WRITE_FILE = 6;

      LITERAL
      FLUSH_LIMIT = 5;

      +
      The following symbols are for the interface to EDT$OPN_OFIDEF. Note that these values
      are hard-coded into the MACRO-11 modules, and into EDT$IOMOD.

      LITERAL
      DISK_FILE_NO = 0,
      DISK_FILE_YES = 1,
      DISK_FILE_RSTS = 2,
      SEQ_NO = 0,
      SEQ_YES = 1,
      RELAT_NONE = 0,
      RELAT_INPUT = 1,
      ATTR_INPUT = 0,
      ATTR_DEFAULT = 1,
      ATTR_JOURNAL = 2;

      OWN STORAGE:
      in the routine

      EXTERNAL REFERENCES:
      in the routine
```

```

: we wish to open a file for both input and output
: code signifying we want to get a record from a file
: code signifying we want to put a record to a file
: we want to close the file and then delete it
: we want to close the file
: code for the startup command file
: code for the main input file
: code for an include file
: code for the journal file
: code for the output file
: code for an output file being written
```

```

: ! Flush the journal file buffer after this many records
```

```

: Not a disk file
: Is a disk file
: Is a disk file on RSTS
: The file is not to be sequenced
: The file is to be sequenced
: There is no related file name
: The primary input file is used as the related file
: Take file attributes from the primary input file
: Use EDT's default file attributes
: Use journal file attributes
```

```
0816 1 %SBTTL 'EDT$FILEIO - Central EDT file I/O routine'
0817 1
0818 1 GLOBAL ROUTINE EDT$FILEIO (
0819 1     FILECODE,
0820 1     FILESTRM,
0821 1     FILE_REC,
0822 1     FILE_RHB
0823 1 ) =
0824 1
0825 1 ++
0826 1 FUNCTIONAL DESCRIPTION:
0827 1
0828 1 This is the basic file I/O routine for EDT. Callable EDT calls this
0829 1 routine to do any I/O if this is the routine passed to it by the calling
0830 1 program. This is the routine passed to callable EDT by the "real" EDT.
0831 1
0832 1 FORMAL PARAMETERS:
0833 1
0834 1 filecode = address of fullword containing function code defining type of I/O
0835 1 operation to be performed
0836 1 filestream = address of fullword containing stream identifier
0837 1 file_rec = address of string descriptor, i.e. the file name or place to store
0838 1 record read or place to fetch record to be written
0839 1 file_rhb = address of string descriptor for any record prefixes
0840 1
0841 1
0842 1 Note: the default name is not implemented for WRITE/EXIT/PRINT files
0843 1 (because of .TMP logic). Fortunately, EDT does not pass a default
0844 1 name on these channels.
0845 1
0846 1 IMPLICIT INPUTS:
0847 1
0848 1 EDT$$Z_SYS_PRIRAB
0849 1 EDT$$Z_SYS_JOURAB
0850 1 EDT$$Z_SYS_CMDRAB
0851 1 EDT$$Z_SYS_ALTRAB
0852 1
0853 1
0854 1 IMPLICIT OUTPUTS:
0855 1
0856 1 EDT$$Z_SYS_PRIRAB
0857 1 EDT$$Z_SYS_JOURAB
0858 1 EDT$$Z_SYS_CMDRAB
0859 1 EDT$$Z_SYS_ALTRAB
0860 1
0861 1 COMPLETION STATUS:
0862 1
0863 1 The only error returned, rather than signaled, is EOF.
0864 1
0865 1 SIDE EFFECTS:
0866 1
0867 1 NONE
0868 1
0869 1 --
0870 1
0871 2 BEGIN
0872 2
```



```

274 0873 2 MAP
275 0874 2 FILE_REC : REF BLOCK [, BYTE],
276 0875 2 FILE_RMB : REF BLOCK [, BYTE];
277 0876 2
278 0877 2 EXTERNAL ROUTINE
279 0878 2 EDT$PAR_FNAME,
280 0879 2 EDT$CNV_UPC,
281 0880 2 EDT$REN_FI,
282 0881 2 EDT$FLUSH_OBUF,
283 0882 2 EDT$OPN_IFIDEF,
284 0883 2 EDT$OPN_OFIDEF,
285 0884 2 EDT$CLS_FI,
286 0885 2 EDT$RD_IFI,
287 0886 2 EDT$WR_OFI;
288 0887 2
289 L 0888 2 %IF %BLISS (BLISS32)
290 0889 2 %THEN
291 0890 2
292 0891 2 EXTERNAL ROUTINE
293 0892 2 STR$FREE1_DX,
294 0893 2 EDT$OPN_INOUT,
295 0894 2 STR$COPY_DX,
296 0895 2 STR$COPY_R;
297 0896 2
298 0897 2 %FI
299 0898 2
300 0899 2 EXTERNAL
301 0900 2 EDT$Z_SYS_PRIAB,
302 0901 2 EDT$Z_SYS_JOURAB,
303 0902 2 EDT$Z_SYS_CMDRAB,
304 0903 2 EDT$Z_SYS_ALTRAB;
305 0904 2
306 0905 2 MESSAGES ((INPFILOPN, FILNAM, INTERERR, COMFILNEX, COMFILNOP, NOJNLFIL, INPFILNEX, OUTFILCRE, NONSTDFIL)
307 0906 2
308 L 0907 2 %IF %BLISS (BLISS32)
309 0908 2 %THEN
310 0909 2 !+
311 0910 2 ! Keep the filename descriptor for each file - on VMS it's a dynamic descriptor
312 0911 2 !-
313 0912 2
314 0913 2 OWN
315 0914 2 CMD_DESC : BLOCK [8, BYTE] ! command file
316 0915 2 -PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
317 0916 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
318 0917 2 [DSC$A_POINTER] = 0,
319 0918 2 [DSC$W_LENGTH] = 0),
320 0919 2 JOU_DESC : BLOCK [8, BYTE] ! journal file
321 0920 2 -PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
322 0921 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
323 0922 2 [DSC$A_POINTER] = 0,
324 0923 2 [DSC$W_LENGTH] = 0),
325 0924 2 INP_DESC : BLOCK [8, BYTE] ! primary input file
326 0925 2 -PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,
327 0926 2 [DSC$B_CLASS] = DSC$K_CLASS_D,
328 0927 2 [DSC$A_POINTER] = 0,
329 0928 2 [DSC$W_LENGTH] = 0),
330 0929 2 ALT_DESC : BLOCK [8, BYTE] ! temporary or secondary file
```

```
331      PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,  
332               [DSC$B_CLASS] = DSC$K_CLASS_D,  
333               [DSC$A_POINTER] = 0,  
334               [DSC$W_LENGTH] = 0),  
335      OUT_DESC : BLOCK [8, BYTE] ! output file  
336      PRESET ( [DSC$B_DTYPE] = DSC$K_DTYPE_T,  
337               [DSC$B_CLASS] = DSC$K_CLASS_D,  
338               [DSC$A_POINTER] = 0,  
339               [DSC$W_LENGTH] = 0);  
340  
341      !+  
342      ! The resultant name from the primary input open, used for the primary output open.  
343      ! (We cannot use INP_DESC since it is released after the input file is closed,  
344      ! which may be before the output file is opened.)  
345      !-  
346  
347      OWN  
348      INP_NAME : VECTOR [256, BYTE],  
349      INP_NAME_LEN;  
350  
351      U 0950      %ELSE  
352      U 0951  
353      U 0952      OWN  
354      U 0953      CMD_DESC : BLOCK [8, BYTE] ! command file  
355      U 0954      PRESET ( [DSC$A_POINTER] = 0,  
356      U 0955      [DSC$W_LENGTH] = 0),  
357      U 0956      JOU_DESC : BLOCK [8, BYTE] ! journal file  
358      U 0957      PRESET ( [DSC$A_POINTER] = 0,  
359      U 0958      [DSC$W_LENGTH] = 0),  
360      U 0959      INP_DESC : BLOCK [8, BYTE] ! main input file  
361      U 0960      PRESET ( [DSC$A_POINTER] = 0,  
362      U 0961      [DSC$W_LENGTH] = 0),  
363      U 0962      ALT_DESC : BLOCK [8, BYTE] ! temporary or secondary file  
364      U 0963      PRESET ( [DSC$A_POINTER] = 0,  
365      U 0964      [DSC$W_LENGTH] = 0),  
366      U 0965      OUT_DESC : BLOCK [8, BYTE] ! output file  
367      U 0966      PRESET ( [DSC$A_POINTER] = 0,  
368      U 0967      [DSC$W_LENGTH] = 0);  
369      U 0968  
370      U 0969      %FI  
371      U 0970  
372      U 0971      OWN  
373      U 0972  
374      L 0973      %IF %BLISS (BLISS32)  
375      U 0974      %THEN  
376      U 0975      OUT_IFI, ! internal file id for primary output file  
377      U 0976      JOU_IFI, ! internal file id for journal file  
378      U 0977      INCL_IFI, ! internal file id for include file  
379      U 0978      INP_IFI, ! internal file id for primary input  
380      U 0979      CMD_IFI, ! internal file id for command file  
381      U 0980      %FI  
382      U 0981  
383      U 0982      DISK_FI, ! flag indicating opening a renameable file for output  
384      U 0983      FLUSH_COUNTER : INITIAL (0), ! counts PUTs to journal towards flushing the buffer  
385      U 0984      INCL_VFC, ! flag indicating include file is VFC format file  
386      U 0985      INPUT_VFC; ! flag indicating primary input is VFC format file  
387      U 0986
```



```

388      LOCAL
389      VFC,
390      ERROR,
391      IO_STS,
392      IO_STV,
393      STATUS;
394
395      BIND
396      FILE_DESC = .FILE_REC : BLOCK [, BYTE], ! passed in descriptor for filename or record in or out
397      RHB_DESC = .FILE_RHB : BLOCK [, BYTE]; ! record header block descriptor
398
399      !+
400      !- Find out first what kind of operation is requested
401
402      CASE .FILECODE FROM EDT$K_OPEN_INPUT TO EDT$K_CLOSE OF
403      SET
404      !+
405      !- Open a file for input
406
407      [EDT$K_OPEN_INPUT] :
408      BEGIN
409      ! we want to open a file
410
411      LOCAL
412      NONSTD;
413
414      L 1014 %IF %BLISS (BLISS16)
415      U 1015 %THEN
416      EDT$CNV_UPC (.FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH]);
417      1016 %FI
418
419      NONSTD = 0;
420
421      CASE .FILESTRM FROM EDT$K_COMMAND_FILE TO EDT$K_INCLUDE_FILE OF
422      SET
423      ! which file?
424
425      [EDT$K_COMMAND_FILE] :
426      BEGIN
427      ! open the command file for input
428
429      L 1027 %IF %BLISS (BLISS32)
430      U 1028 %THEN
431      CMD_IFI = EDT$OPN_IFIDEF (EDT$Z_SYS_CMDRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
432      .RHB_DESC [DSC$W_LENGTH], RELAT_NONE, IO_STS, IO_STV, VFC, NONSTD);
433
434      !+
435      !- If the open failed then find out why
436
437      IF (.CMD_IFI EQL 0)
438      THEN
439      !+
440      !- Signal an error
441
442      SIGNAL_STOP (SHR$OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, FILE_DESC,
443      .IO_STS, .IO_STV);
444
445      !+

```

```

445      1044 4 ! If the file is non-standard, indicate this.
446      1045 4 !-
447      1046 4
448      1047 4
449      1048 4 IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
450      1049 4
451      1050 4 !+ Save the complete filename
452      1051 4 !-
453      1052 4 STRING_DESC (CMD_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
454      U 1053 4 %ELSE
455      U 1054 4 IO_STS = EDT$OPN_IFIDEF (EDT$$Z SYS CMDRAB, .FILE_DESC [DSC$A_POINTER],
456      U 1055 4 .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH], 0, 0
457      U 1056 4 0, 0);
458      1057 4 %FI
459      1058 4
460      1059 4 RETURN (.IO_STS); ! return status
461      1060 3 END;
462      1061 3
463      1062 3 [EDT$K_INPUT_FILE] : ! open the primary input file for input
464      1063 4 BEGIN
465      1064 4
466      L 1065 4 %IF %BLISS (BLISS32)
467      1066 4 %THEN
468      1067 4 INP_IFI = EDT$OPN_IFIDEF (EDT$$Z SYS PRIRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
469      1068 4 .RHB_DESC [DSC$W_LENGTH], RELAT_NONE, IO_STS, IO_STV, INPUT_VFC, NONSTD);
470      1069 4 !+
471      1070 4 ! Save the name for opening the output file on VMS, even if the input file does not open.
472      1071 4 !-
473      1072 4 INP_NAME_LEN = .FILE_DESC [DSC$W_LENGTH];
474      1073 4 CH$MOVE (INP_NAME_LEN, .FILE_DESC [DSC$A_POINTER], INP_NAME);
475      1074 4 !+
476      1075 4 ! Check for open failure.
477      1076 4 !-
478      1077 4
479      1078 5 IF (.INP_IFI EQL 0)
480      1079 4 THEN
481      1080 4 SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
482      1081 4 1, FILE_DESC, .IO_STS, .IO_STV);
483      1082 4
484      1083 4 !+
485      1084 4 ! If the file is non-standard, indicate this.
486      1085 4 !-
487      1086 4
488      1087 4 IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
489      1088 4
490      U 1089 4 %ELSE
491      U 1090 4 IO_STS = EDT$OPN_IFIDEF (EDT$$Z SYS PRIRAB, .FILE_DESC [DSC$A_POINTER],
492      U 1091 4 .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH], 0, 0
493      U 1092 4 0, 0);
494      1093 4 %FI
495      1094 4
496      1095 4 !+
497      1096 4 ! Save the complete filename. This is needed on the PDP-11 for opening the journal file.
498      1097 4 !-
499      1098 4 STRING_DESC (INP_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
500      1099 4 RETURN (.IO_STS); ! return status
501      1100 3 END;
```



```

502      1101 3
503      1102 3          [EDT$K_INCLUDE_FILE] :          ! open include file for input
504      1103 4          BEGIN
505      1104 4
506      L 1105 4 %IF %BLISS (BLISS32)
507      1106 4 %THEN
508      1107 5          BEGIN
509      1108 5          INCL_IFI = EDT$$OPN_IFIDEF (EDT$$Z_SYS_ALTRAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
510      1109 5          .RHB_DESC [DSC$W_LENGTH], RELAT_INPUT, IO_STS, IO_STV, INCL_VFC, NONSTD);
511      1110 5
512      1111 6          IF (.INCL_IFI EQL 0)
513      1112 5          THEN
514      1113 5          !+
515      1114 5          !- Signal the error
516      1115 5
517      1116 5          SIGNAL_STOP (SHR$_OPENIN + (EDT$K_FAC_NO*65536) + ST$K_SEVERE, 1, FILE_DESC,
518      1117 5          .IO_STS, .IO_STV);
519      1118 5
520      1119 5          !+
521      1120 5          !- If the file is non-standard, indicate this.
522      1121 5
523      1122 5
524      1123 5          IF .NONSTD THEN IO_STS = EDT$_NONSTDFIL;
525      1124 5
526      1125 5          !+
527      1126 5          !- Save the complete filename
528      1127 5
529      1128 5          STRING_DESC (ALT_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
530      1129 4          END;
531      U 1130 4 %ELSE
532      U 1131 4          IO_STS = EDT$$OPN_IFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
533      U 1132 4          .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
534      U 1133 4          .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 0, 0);
535      1134 4 %FI
536      1135 4
537      1136 4          RETURN (.IO_STS);
538      1137 3          END;
539      1138 3
540      1139 3          [INRANGE, OTRANGE] :
541      1140 3          ASSERT (0);
542      1141 3          TES;
543      1142 3
544      1143 2          END;
545      1144 2          !+
546      1145 2          !- Open a file for output
547      1146 2
548      1147 2
549      1148 2          [EDT$K_OPEN_OUTPUT_SEQ, EDT$K_OPEN_OUTPUT_NOSEQ] :
550      1149 3          BEGIN
551      1150 3
552      1151 3          LOCAL
553      1152 3          SEQ;
554      1153 3
555      L 1154 3          %IF %BLISS (BLISS16)
556      U 1155 3          %THEN
557      U 1156 3          EDT$$CNV_UPC (.FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH]);
558      1157 3          %FI
```

```

559      1158 3
560      1159 4
561      1160 3
562      1161 3
563      1162 3
564      1163 3
565      1164 3
566      1165 3
567      1166 3
568      1167 3
569      1168 3
570      1169 4
571      1170 4
572      1171 4
573      1172 4
574      1173 4
575      1174 4
576      1175 4
577      1176 5
578      1177 4
579      1178 5
580      1179 5
581      1180 5
582      1181 5
583      1182 5
584      1183 4
585      1184 5
586      1185 5
587      1186 5
588      1187 5
589      1188 4
590      1189 4
591      1190 4
592      1191 4
593      1192 4
594      1193 4
595      1194 4
596      1195 4
597      L 1196 4
598      1197 4
599      1198 4
600      1199 4
601      1200 4
602      1201 4
603      1202 4
604      1203 4
605      1204 5
606      1205 4
607      1206 5
608      1207 5
609      1208 4
610      1209 4
611      1210 4
612      1211 4
613      1212 4
614      1213 4
615      1214 4

      IF (...FILECODE EQL EDT$K_OPEN_OUTPUT_SEQ)
      THEN
          SEQ = SEQ_YES                ! make it a sequenced VFC file
      ELSE
          SEQ = SEQ_NO;                ! not a sequenced file
      CASE ...FILESTRM FROM EDT$K_JOURNAL_FILE TO EDT$K_WRITE_FILE OF
          SET
              [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :      ! WRITE or OUTPUT file
              BEGIN
                  LOCAL
                      ATT,      ! 0 = use input file attributes, 1 = use EDT's default file attributes
                      RELAT,    ! 0 = no related file, 1 = use input file's name and type before default nam
                      FORCE_MAXV; ! 1 = force maximum version number
                  IF (...FILESTRM EQL EDT$K_OUTPUT_FILE)
                  THEN
                      BEGIN
                          ATT = ATTR_INPUT;
                          RELAT = RELAT_INPUT;
                          FORCE_MAXV = T;
                      END
                  ELSE
                      BEGIN
                          ATT = ATTR_DEFAULT;
                          RELAT = RELAT_NONE;
                          FORCE_MAXV = 0;
                      END;
              END;
          !+ This code cannot handle default file names, so make sure there isn't one.
          !-
          ASSERT (.RHB_DESC [DSC$W_LENGTH] EQL 0);
          DISK_FI = 0;
      L 1196 4 %IF %BLISS (BLISS32)
      1197 4 %THEN
      1198 4 !+
      1199 4 ! On VMS, if the EXIT file name is not specified, use the resultant file name from the input open.
      1200 4 ! Because we are forcing maximum version number the version number in the input file name string
      1201 4 ! won't cause trouble.
      1202 4 !-
      1203 4
      1204 5 IF ((.RELAT EQL RELAT_INPUT) AND (.FILE_DESC [DSC$W_LENGTH] EQLU 0))
      1205 4 THEN
      1206 5 BEGIN
      1207 5 STRING_DESC (FILE_DESC, INP_NAME_LEN, INP_NAME);
      1208 4 END;
      1209 4
      1210 4 !+
      1211 4 ! Parse the output file name - If successful, then do the open; otherwise
      1212 4 ! signal an error on open
      1213 4 !-
      1214 4 STATUS = EDT$$PAR_FNAME (EDT$$Z_SYS_ALTRAB, FILE_DESC, .RELAT, DISK_FI, IO_STS, IO_STV);
```



```

: 616      1215  4
: 617      1216  5          IF ( NOT .STATUS)
: 618      1217  4          THEN
: 619      1218  4              SIGNAL_STOP (SHR$_OPENOUT + (EDT$_FAC_NO*65536) + STS$_SEVERE,
: 620      1219  4              1, FILE_DESC, .IO_STS, .IO_STV);
: 621      1220  4
: 622      1221  4              OUT_DESC [DSC$_LENGTH] = 0;
: 623      1222  4              OUT_DESC [DSC$_POINTER] = 0;
: 624      1223  4      !+
: 625      1224  4      ! Save description of output file before translation with .TMP extension
: 626      1225  4      ! if this is a disk or DEctape file for rename later
: 627      1226  4      !-
: 628      1227  4
: 629      1228  5          IF (.DISK_FI)
: 630      1229  4          THEN
: 631      1230  5              BEGIN
: 632      1231  5                  STRING_DESC (OUT_DESC, FILE_DESC [DSC$_LENGTH], .FILE_DESC [DSC$_POINTER]);
: 633      1232  5                  STR$COPY R (FILE_DESC, %REF (%CHARCOUNT (TEMP_TYP)), UPLIT (BYTE (TEMP_TYP)));
: 634      1233  5                  FORCE_MAXV = 1;          ! For .TMP file, force max version number
: 635      1234  4                  END;
: 636      1235  4
: 637      1236  4      !+
: 638      1237  4      ! If this is a disk file, open a temporary file for output, then rename later
: 639      1238  4      ! if all goes well.  If not a disk file, just open the "given" file.
: 640      1239  4      !-
: 641      1240  4          OUT_IFI = EDT$_OPN OFIDEF (EDT$_SYS_ALTRAB, FILE_DESC, .OUT_DESC [DSC$_POINTER],
: 642      1241  4          .OUT_DESC [DSC$_LENGTH], .SEQ, .RELAT, .ATT, .FORCE_MAXV, IO_STS, IO_STV);
: 643      1242  4      !+
: 644      1243  4      ! Signal an error
: 645      1244  4      !-
: 646      1245  4
: 647      1246  5          IF (.OUT_IFI EQL 0)
: 648      1247  4          THEN
: 649      1248  4              SIGNAL_STOP (SHR$_OPENOUT + (EDT$_FAC_NO*65536) + STS$_SEVERE,
: 650      1249  4              1, FILE_DESC, .IO_STS, .IO_STV);
: 651      1250  4
: 652      1251  4      !+
: 653      1252  4      ! Save the complete filename for the close later
: 654      1253  4      !-
: 655      1254  4          STRING_DESC (ALT_DESC, FILE_DESC [DSC$_LENGTH], .FILE_DESC [DSC$_POINTER]);
: 656      U 1255  4      %ELSE
: 657      U 1256  4
: 658      U 1257  4          IF (.RELAT EQL RELAT_INPUT)
: 659      U 1258  4          THEN
: 660      U 1259  4              BEGIN
: 661      U 1260  4                  STATUS = EDT$_PAR FNAME (EDT$_SYS_ALTRAB, .FILE_DESC [DSC$_POINTER],
: 662      U 1261  4                  .FILE_DESC [DSC$_LENGTH], .INP_DESC [DSC$_POINTER], .INF_DESC [DSC$_LENGTH],
: 663      U 1262  4                  DISK_FI);
: 664      U 1263  4              END
: 665      U 1264  4          ELSE
: 666      U 1265  4              BEGIN
: 667      U 1266  4                  STATUS = EDT$_PAR FNAME (EDT$_SYS_ALTRAB, .FILE_DESC [DSC$_POINTER],
: 668      U 1267  4                  .FILE_DESC [DSC$_LENGTH], 0, 0, DISK_FI);
: 669      U 1268  4              END;
: 670      U 1269  4
: 671      U 1270  4          STRING_DESC (OUT_DESC, FILE_DESC [DSC$_LENGTH], .FILE_DESC [DSC$_POINTER]);
: 672      U 1271  4
```

```

673      U 1272 4      IF (.STATUS)
674      U 1273 4      THEN
675      U 1274 4      !+
676      U 1275 4      Disk files are handled specially on RSTS. We don't use a .TMP extension
677      U 1276 4      but rather open it in temporary mode using the actual name given
678      U 1277 4      !-
679      U 1278 4
680      U 1279 4      IF (.DISK_FI EQL DISK_FILE_YES)
681      U 1280 4      THEN
682      U 1281 4      BEGIN
683      U 1282 4
684      U 1283 4      IF (.RELAT EQL RELAT_INPUT)
685      U 1284 4      THEN
686      U 1285 4      BEGIN
687      U 1286 4      IO_STS = EDT$$OPN OFIDEF (EDT$$Z_SYS_ALTRAB, UPLIT (BYTE (TEMP_TYP)),
688      U 1287 4      %CHARCOUNT (TEMP_TYP), .FILE_DESC [DSC$A_POINTER],
689      U 1288 4      .FILE_DESC [DSC$W_LENGTH], .INP_DESC [DSC$A_POINTER],
690      U 1289 4      .INP_DESC [DSC$W_LENGTH], 1, 0, .SEQ, .ATT);
691      U 1290 4      END
692      U 1291 4      ELSE
693      U 1292 4      BEGIN
694      U 1293 4      IO_STS = EDT$$OPN OFIDEF (EDT$$Z_SYS_ALTRAB, UPLIT (BYTE (TEMP_TYP)),
695      U 1294 4      %CHARCOUNT (TEMP_TYP), .FILE_DESC [DSC$A_POINTER],
696      U 1295 4      .FILE_DESC [DSC$W_LENGTH], 0, 0, 1, 0, .SEQ, .ATT);
697      U 1296 4      END;
698      U 1297 4
699      U 1298 4      END
700      U 1299 4      ELSE
701      U 1300 4      BEGIN
702      U 1301 4
703      U 1302 4      IF (.RELAT EQL RELAT_INPUT)
704      U 1303 4      THEN
705      U 1304 4      BEGIN
706      U 1305 4      IO_STS = EDT$$OPN OFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
707      U 1306 4      .FILE_DESC [DSC$W_LENGTH], 0, 0, .INP_DESC [DSC$A_POINTER],
708      U 1307 4      .INP_DESC [DSC$W_LENGTH], .FORCE_MAXV, 0, .SEQ, .ATT);
709      U 1308 4      END
710      U 1309 4      ELSE
711      U 1310 4      BEGIN
712      U 1311 4      IO_STS = EDT$$OPN OFIDEF (EDT$$Z_SYS_ALTRAB, .FILE_DESC [DSC$A_POINTER],
713      U 1312 4      .FILE_DESC [DSC$W_LENGTH], 0, 0, 0, 0, .FORCE_MAXV, 0, .SEQ, .ATT);
714      U 1313 4      END;
715      U 1314 4
716      U 1315 4      END
717      U 1316 4      ELSE
718      U 1317 4      IO_STS = .STATUS;
719      U 1318 4
720      U 1319 4      XFI
721      U 1320 4
722      U 1321 4      RETURN (.IO_STS);
723      U 1322 4      END;
724      U 1323 3
725      U 1324 3      [EDT$K JOURNAL_FILE] :
726      U 1325 3      BEGIN
727      U 1326 4
728      U 1327 4
729      L 1328 4      XIF XBLISS (BLISS32)
```



```

730      1329 4 %THEN
731      1330 4          JOU_IFI = EDT$OPN OFIDEF (EDT$$Z_SYS_JOURAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
732      1331 4          .RHB_DESC [DSC$W_LENGTH], SEQ_NO, RELAT_INPUT, ATTR_JOURNAL, 1, IO_STS, IO_STV);
733      1332 4
734      1333 5          IF (.JOU_IFI EQL 0)
735      1334 4          THEN
736      1335 4              SIGNAL_STOP (SHR$ OPENOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE,
737      1336 4              1, FILE_DESC, .IO_STS, .IO_STV);
738      1337 4
739      1338 4              STRING_DESC (JOU_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
740      1339 4 %ELSE
741      1340 4 !+
742      1341 4 !- Note that .SEQ+1 is used to specify a normal output open or an open for append.
743      1342 4 !-
744      1343 4          IO_STS = EDT$OPN OFIDEF (EDT$$Z_SYS_JOURAB, .FILE_DESC [DSC$A_POINTER],
745      1344 4          .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
746      1345 4          .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 1, .SEQ + 1, 0, 1);
747      1346 4 %FI
748      1347 4
749      1348 4          RETURN (.IO_STS);
750      1349 3          END;
751      1350 3
752      1351 3          [INRANGE, OUTRANGE] :
753      1352 3          ASSERT (0);
754      1353 3          TES;
755      1354 3
756      1355 2          END;
757      1356 2 !+
758      1357 2 !- Open a file for both input and output
759      1358 2 !-
760      1359 2
761      1360 2          [EDT$K_OPEN_IN_OUT] :
762      1361 3          BEGIN
763      1362 3 !+
764      1363 3 !- The journal file is the only file we can open this way
765      1364 3 !-
766      1365 3
767      1366 4          IF (..FILESTRM EQL EDT$K_JOURNAL_FILE)
768      1367 3          THEN
769      1368 4          BEGIN
770      1369 4
771      1370 4 %IF %BLISS (BLISS32)
772      1371 4 %THEN
773      1372 4          JOU_IFI = EDT$OPN INOUT (EDT$$Z_SYS_JOURAB, FILE_DESC, .RHB_DESC [DSC$A_POINTER],
774      1373 4          .RHB_DESC [DSC$W_LENGTH], IO_STS, IO_STV);
775      1374 4
776      1375 5          IF (.JOU_IFI EQL 0)
777      1376 4          THEN
778      1377 4              SIGNAL_STOP (SHR$ OPENIN + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1,
779      1378 4              FILE_DESC, .IO_STS, .IO_STV);
780      1379 4
781      1380 4              STRING_DESC (JOU_DESC, FILE_DESC [DSC$W_LENGTH], .FILE_DESC [DSC$A_POINTER]);
782      1381 4 %ELSE
783      1382 4          IO_STS = EDT$OPN IFIDEF (EDT$$Z_SYS_JOURAB, .FILE_DESC [DSC$A_POINTER],
784      1383 4          .FILE_DESC [DSC$W_LENGTH], .RHB_DESC [DSC$A_POINTER], .RHB_DESC [DSC$W_LENGTH],
785      1384 4          .INP_DESC [DSC$A_POINTER], .INP_DESC [DSC$W_LENGTH], 0, 1);
786      1385 4 %FI
```

```

: 787      1386 4
: 788      1387 4
: 789      1388 4
: 790      1389 3
: 791      1390 3
: 792      1391 3
: 793      1392 2
: 794      1393 2
: 795      1394 2
: 796      1395 3
: 797      1396 3
: 798      1397 3
: 799      1398 3
: 800      1399 3
: 801      1400 3
: 802      1401 3
: 803      1402 3
: 804      1403 3
: 805      1404 3
: 806      1405 4
: 807      1406 4
: 808      1407 4
: 809      1408 4
: 810      1409 3
: 811      1410 3
: 812      1411 3
: 813      1412 4
: 814      1413 4
: 815      1414 4
: 816      1415 4
: 817      1416 3
: 818      1417 3
: 819      1418 3
: 820      1419 4
: 821      1420 4
: 822      1421 4
: 823      1422 4
: 824      1423 3
: 825      1424 3
: 826      1425 3
: 827      1426 4
: 828      1427 4
: 829      1428 4
: 830      1429 4
: 831      1430 3
: 832      1431 3
: 833      1432 3
: 834      1433 3
: 835      1434 3
: 836      1435 3
: 837      1436 3
: 838      1437 3
: 839      1438 3
: 840      1439 3
: 841      1440 4
: 842      1441 3
: 843      1442 3

      RETURN (.IO_STS);
      END
      ELSE
      ASSERT (0);
      END;
[EDT$K_GET] :
      BEGIN
      LOCAL
      DESC_ADDR,
      RAB;
      CASE .FILESTRM FROM EDT$K_COMMAND_FILE TO EDT$K_JOURNAL_FILE OF
      SET
      [EDT$K_COMMAND_FILE] :
      ! the startup command file
      BEGIN
      DESC_ADDR = CMD_DESC;
      RAB = EDT$$Z_SYS_CMDRAB;
      VFC = 0;
      END;
      [EDT$K_INPUT_FILE] :
      ! get a record from the primary input file
      BEGIN
      DESC_ADDR = INP_DESC;
      VFC = .INPUT_VFC;
      RAB = EDT$$Z_SYS_PRIIRAB;
      END;
      [EDT$K_INCLUDE_FILE] :
      ! the secondary input file
      BEGIN
      VFC = .INCL_VFC;
      DESC_ADDR = ALT_DESC;
      RAB = EDT$$Z_SYS_ALIRAB;
      END;
      [EDT$K_JOURNAL_FILE] :
      ! get a record from the journal file
      BEGIN
      VFC = 0;
      DESC_ADDR = JOU_DESC;
      RAB = EDT$$Z_SYS_JOURAB;
      END;
      [INRANGE, OUTRANGE] :
      ASSERT (0);
      TES;
      L 1436 %IF %BLISS (BLISS32)
      1437 %THEN
      1438 STATUS = EDT$$RD_IF1 (.RAB, FILE_DESC, RHB_DESC, IO_STS, IO_STV, .VFC);
      1439
      1440 IF ( NOT .STATUS)
      1441 THEN
      1442
```



```

: 844      1443      4      IF (.IO_STS EQL RMS$_EOF)
: 845      1444      3      THEN
: 846      1445      4      RETURN (.IO_STS)
: 847      1446      3      ELSE
: 848      1447      3      SIGNAL_STOP (SHR$_READERR + (EDT$_FAC_NO*65536) + STS$_SEVERE, 1, .DESC_ADDR, .IO_STS,
: 849      1448      3      .IO_STV);
: 850      1449      3
: 851      1450      3      %ELSE
: 852      1451      3      BEGIN
: 853      1452      3
: 854      1453      3      LOCAL
: 855      1454      3      REC_ADDR,
: 856      1455      3      REC_LEN;
: 857      1456      3
: 858      1457      3      STATUS = EDT$_RD IFI (.RAB, REC_ADDR, REC_LEN, .RHB_DESC [DSC$_POINTER], !
: 859      1458      3      RHB_DESC [DSC$_LENGTH]);
: 860      1459      3      STRING_DESC (FILE_DESC, REC_LEN, .REC_ADDR);
: 861      1460      3      END;
: 862      1461      3      %FI
: 863      1462      3
: 864      1463      3      RETURN (.STATUS);
: 865      1464      3      END;
: 866      1465      3
: 867      1466      3      [EDT$_PUT] : ! we wish to put a record to a file
: 868      1467      3      BEGIN
: 869      1468      3
: 870      1469      3      LOCAL
: 871      1470      3      DESC_ADDR,
: 872      1471      3      RAB;
: 873      1472      3
: 874      1473      3      CASE ..FILESTRM FROM EDT$_JOURNAL_FILE TO EDT$_WRITE_FILE OF
: 875      1474      3      SET
: 876      1475      3
: 877      1476      3      [EDT$_OUTPUT_FILE, EDT$_WRITE_FILE] : ! put a record in an output file
: 878      1477      4      BEGIN
: 879      1478      4      DESC_ADDR = ALT_DESC;
: 880      1479      4      RAB = EDT$_Z_SYS_ALTRAB;
: 881      1480      4      END;
: 882      1481      3
: 883      1482      3      [EDT$_JOURNAL_FILE] : ! put a record to the journal file
: 884      1483      4      BEGIN
: 885      1484      4      DESC_ADDR = JOURNAL_DESC;
: 886      1485      4      RAB = EDT$_Z_SYS_JOURAB;
: 887      1486      4      END;
: 888      1487      3
: 889      1488      3      [INRANGE, OUTRANGE] :
: 890      1489      3      ASSERT (0);
: 891      1490      3      TES;
: 892      1491      3
: 893      1492      3      %IF %BLISS (BLISS32)
: 894      1493      3      %THEN
: 895      1494      3      STATUS = EDT$_WR OFI (.RAB, FILE_DESC, RHB_DESC, IO_STS, IO_STV);
: 896      1495      3      %ELSE
: 897      1496      3      STATUS = EDT$_WR OFI (.RAB, .FILE_DESC [DSC$_POINTER], .FILE_DESC [DSC$_LENGTH],
: 898      1497      3      .RHB_DESC [DSC$_POINTER]);
: 899      1498      3      %FI
: 900      1499      3
```

```

: 901      1500      4      IF ( NOT .STATUS)
: 902      1501      3      THEN
: 903      1502      3
: 904      1503      3      %IF %BLISS (BLISS32)
: 905      1504      3      %THEN
: 906      1505      3      SIGNAL_STOP (SHR$_WRITEERR + (EDT$_FAC_NO*65536) + STS$_SEVERE, 1, .DESC_ADDR, .IO_STS,
: 907      1506      3      .IO_STV)
: 908      1507      3      %FI
: 909      1508      3
: 910      1509      3      ELSE
: 911      1510      3
: 912      1511      4      IF (..FILESTRM EQL EDT$_JOURNAL_FILE)
: 913      1512      3      THEN      ! keep the journal buffer clear
: 914      1513      3      BEGIN
: 915      1514      4      FLUSH_COUNTER = .FLUSH_COUNTER + 1;
: 916      1515      4
: 917      1516      5      IF (.FLUSH_COUNTER EQL FLUSH_LIMIT)
: 918      1517      4      THEN
: 919      1518      5      BEGIN
: 920      1519      5
: 921      1520      5      %IF %BLISS (BLISS32)
: 922      1521      5      %THEN
: 923      1522      5      STATUS = EDT$$FLUSH_OBUF (.RAB, IO_STV);
: 924      1523      5
: 925      1524      6      IF ( NOT .STATUS)
: 926      1525      5      THEN
: 927      1526      5      SIGNAL_STOP (SHR$_WRITEERR + (EDT$_FAC_NO*65536) + STS$_SEVERE, 1, .DESC_ADDR,
: 928      1527      5      .STATUS, .IO_STV);
: 929      1528      5
: 930      1529      5      %ELSE
: 931      1530      5      STATUS = EDT$$FLUSH_OBUF (.RAB);
: 932      1531      5      %FI
: 933      1532      5
: 934      1533      5      FLUSH_COUNTER = 0;
: 935      1534      4      END;
: 936      1535      4
: 937      1536      3      END;
: 938      1537      3
: 939      1538      3      RETURN (.STATUS);
: 940      1539      3      END;
: 941      1540      3
: 942      1541      3      [EDT$_CLOSE] :      ! close a file
: 943      1542      3      BEGIN
: 944      1543      3
: 945      1544      3      LOCAL
: 946      1545      3      DESC_ADDR,
: 947      1546      3      ERROR;
: 948      1547      3
: 949      1548      3      CASE .FILESTRM FROM EDT$_COMMAND_FILE TO EDT$_WRITE_FILE OF
: 950      1549      3      SET
: 951      1550      3
: 952      1551      3      [EDT$_COMMAND_FILE] :      ! close the command file
: 953      1552      4      BEGIN
: 954      1553      4
: 955      1554      4      %IF %BLISS (BLISS32)
: 956      1555      4      %THEN
: 957      1556      4      DESC_ADDR = CMD_DESC;
```



```

: 958      1557 4      ERROR = SHR$ CLOSEIN;
: 959      1558 4      EDT$$CLS_FI 7.CMD_IFI, EDT$$Z_SYS_CMDRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
: 960      1559 4 %ELSE
: 961      1560 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_CMDRAB, 0);
: 962      1561 4 %FI
: 963      1562 4
: 964      1563 3      END;
: 965      1564 3
: 966      1565 3      [EDT$K_INPUT_FILE] :      ! close the primary input ifle
: 967      1566 4      BEGIN
: 968      1567 4
: 969      L 1568 4 %IF %BLISS (BLISS32)
: 970      1569 4 %THEN
: 971      1570 4      DESC_ADDR = INP_DESC;
: 972      1571 4      ERROR = SHR$ CLOSEIN;
: 973      1572 4      EDT$$CLS_FI 7.INP_IFI, EDT$$Z_SYS_PRIAB, 0, .DESC_ADDR, IO_STS, IO_STV);
: 974      1573 4 %ELSE
: 975      1574 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_PRIAB, 0);
: 976      1575 4 %FI
: 977      1576 4
: 978      1577 3      END;
: 979      1578 3
: 980      1579 3      [EDT$K_INCLUDE_FILE] :      ! close the secondary input file
: 981      1580 4      BEGIN
: 982      1581 4
: 983      L 1582 4 %IF %BLISS (BLISS32)
: 984      1583 4 %THEN
: 985      1584 4      DESC_ADDR = ALT_DESC;
: 986      1585 4      EDT$$CLS_FI (.INCL_IFI, EDT$$Z_SYS_ALTRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
: 987      1586 4      ERROR = SHR$ CLOSEIN;
: 988      1587 4 %ELSE
: 989      1588 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
: 990      1589 4 %FI
: 991      1590 4
: 992      1591 3      END;
: 993      1592 3
: 994      1593 3      [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :      ! close an output file
: 995      1594 4      BEGIN
: 996      1595 4
: 997      1596 4      LOCAL
: 998      1597 4      FORCE_MAXV;
: 999      1598 4
: 1000     1599 4      IF (..FILESTRM EQL EDT$K_OUTPUT_FILE) THEN FORCE_MAXV = 1 ELSE FORCE_MAXV = 0;
: 1001     1600 4
: 1002     L 1601 4 %IF %BLISS (BLISS32)
: 1003     1602 4 %THEN
: 1004     1603 4      DESC_ADDR = ALT_DESC;
: 1005     1604 4      ERROR = SHR$ CLOSEOUT;
: 1006     1605 4      EDT$$CLS_FI 7.OUT_IFI, EDT$$Z_SYS_ALTRAB, 0, .DESC_ADDR, IO_STS, IO_STV);
: 1007     1606 4      !+
: 1008     1607 4      !- Check the status from the close
: 1009     1608 4      !-
: 1010     1609 4
: 1011     1610 5      IF (.IO_STS)
: 1012     1611 4      THEN
: 1013     1612 4
: 1014     1613 5      IF (.DISK_FI)
```



```

: 1015      1614 4      THEN
: 1016      1615 5      BEGIN
: 1017      1616 5      EDT$$REN FI (ALT_DESC, OUT_DESC, .FORCE_MAXV, IO_STS, IO_STV);
: 1018      1617 5      STRING_DESC (FILE_DESC, OUT_DESC [DSC$W_LENGTH], .OUT_DESC [DSC$A_POINTER]);
: 1019      1618 5      END
: 1020      1619 4      ELSE
: 1021      1620 4      STRING_DESC (FILE_DESC, ALT_DESC [DSC$W_LENGTH], .ALT_DESC [DSC$A_POINTER]);
: 1022      1621 4
: 1023      U 1622 4 %ELSE
: 1024      U 1623 4
: 1025      U 1624 4      IF (.DISK_FI NEQ DISK_FILE_RSTS) THEN IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
: 1026      U 1625 4
: 1027      U 1626 4      !+
: 1028      U 1627 4      ! If this is a disk file and we had a successful close, then rename the
: 1029      U 1628 4      ! temp file to the name originally given
: 1030      U 1629 4      !-
: 1031      U 1630 4
: 1032      U 1631 4      IF ((.IO_STS) AND (.DISK_FI EQL DISK_FILE_YES))
: 1033      U 1632 4      THEN
: 1034      U 1633 4      IO_STS = EDT$$REN_FI (EDT$$Z_SYS_ALTRAB, .OUT_DESC [DSC$A_POINTER],
: 1035      U 1634 4      .OUT_DESC [DSC$W_LENGTH], .FORCE_MAXV);
: 1036      U 1635 4
: 1037      U 1636 4      !+
: 1038      U 1637 4      ! If this is a RSTS disk file then do a rename of any currently existing
: 1039      U 1638 4      ! files with the originally given name to the same name with a .BAK
: 1040      U 1639 4      ! extension and close the tentative output file making it permanent
: 1041      U 1640 4      !-
: 1042      U 1641 4
: 1043      U 1642 4      IF (.DISK_FI EQL DISK_FILE_RSTS)
: 1044      U 1643 4      THEN
: 1045      U 1644 4      BEGIN
: 1046      U 1645 4      IO_STS = EDT$$REN_FI (EDT$$Z_SYS_ALTRAB, .OUT_DESC [DSC$A_POINTER],
: 1047      U 1646 4      .OUT_DESC [DSC$W_LENGTH], .FORCE_MAXV);
: 1048      U 1647 4
: 1049      U 1648 4      IF (.IO_STS) THEN IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 0);
: 1050      U 1649 4
: 1051      U 1650 4      END;
: 1052      U 1651 4
: 1053      1652 4 %FI
: 1054      1653 4
: 1055      1654 3      END;
: 1056      1655 3
: 1057      1656 3      [EDT$K_JOURNAL_FILE] :      ! close the journal file
: 1058      1657 4      BEGIN
: 1059      1658 4
: 1060      L 1659 4 %IF %BLISS (BLISS32)
: 1061      1660 4 %THEN
: 1062      1661 4      DESC_ADDR = JOURNAL_DESC;
: 1063      1662 4      ERROR = SHR$CLOSEOUT;
: 1064      1663 4      EDT$$CLS_FI (.JOURNAL_FI, EDT$$Z_SYS_JOURNAL, 0, .DESC_ADDR, IO_STS, IO_STV);
: 1065      U 1664 4 %ELSE
: 1066      U 1665 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_JOURNAL, 0);
: 1067      1666 4 %FI
: 1068      1667 4
: 1069      1668 3      END;
: 1070      1669 3
: 1071      1670 3      [INRANGE, OUTRANGE] :
```



```
: 1072      1671 3      ASSERT (0);
: 1073      1672 3      TES;
: 1074      1673 3
: 1075      L 1674 3      %IF %BLISS (BLISS32)
: 1076      1675 3      %THEN
: 1077      1676 3      +
: 1078      1677 3      | Check the status from either the close or the rename of output files
: 1079      1678 3      | -
: 1080      1679 3
: 1081      1680 3      IF ( NOT .IO_STS)
: 1082      1681 3      THEN
: 1083      1682 3      SIGNAL_STOP (.ERROR + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1, .DESC_ADDR,
: 1084      1683 3      .IO_STS, .IO_STV);
: 1085      1684 3
: 1086      1685 3      STR$FREE1_DX (.DESC_ADDR);
: 1087      1686 3      %FI
: 1088      1687 3
: 1089      1688 3      RETURN (.IO_STS);
: 1090      1689 3      END;
: 1091      1690 3
: 1092      1691 3      [EDT$K_CLOSE_DEL] :
: 1093      1692 3      BEGIN
: 1094      1693 3
: 1095      1694 3      LOCAL
: 1096      1695 3      DESC_ADDR;
: 1097      1696 3
: 1098      1697 3      CASE ..FILESTRM FROM EDT$K_JOURNAL_FILE TO EDT$K_WRITE_FILE OF
: 1099      1698 3      SET
: 1100      1699 3
: 1101      1700 3      [EDT$K_OUTPUT_FILE, EDT$K_WRITE_FILE] :
: 1102      1701 4      BEGIN
: 1103      1702 4
: 1104      L 1703 4      %IF %BLISS (BLISS32)
: 1105      1704 4      %THEN
: 1106      1705 4      DESC_ADDR = ALT_DESC;
: 1107      1706 4      EDT$$CLS_FI (.OUT_IFI, EDT$$Z_SYS_ALTRAB, 1, ALT_DESC, IO_STS, IO_STV);
: 1108      U 1707 4      %ELSE
: 1109      U 1708 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_ALTRAB, 1);
: 1110      1709 4      %FI
: 1111      1710 4
: 1112      1711 3      END;
: 1113      1712 3
: 1114      1713 3      [EDT$K_JOURNAL_FILE] :
: 1115      1714 4      BEGIN
: 1116      1715 4
: 1117      L 1716 4      %IF %BLISS (BLISS32)
: 1118      1717 4      %THEN
: 1119      1718 4      DESC_ADDR = JOU_DESC;
: 1120      1719 4      EDT$$CLS_FI (.JOU_IFI, EDT$$Z_SYS_JOURAB, 2, JOU_DESC, IO_STS, IO_STV);
: 1121      U 1720 4      %ELSE
: 1122      U 1721 4      IO_STS = EDT$$CLS_FI (EDT$$Z_SYS_JOURAB, 2);
: 1123      1722 4      %FI
: 1124      1723 4
: 1125      1724 3      END;
: 1126      1725 3
: 1127      1726 3      [INRANGE, OUTRANGE] :
: 1128      1727 3      ASSERT (0);
```

```
: 1129      1728      3      TES;
: 1130      1729      3
: 1131      1730      3      %IF %BLISS (BLISS32)
: 1132      1731      3      %THEN
: 1133      1732      3
: 1134      1733      4      IF ( NOT .IO_STS)
: 1135      1734      3      THEN
: 1136      1735      3      SIGNAL_STOP (SHR$_CLOSEOUT + (EDT$K_FAC_NO*65536) + STS$K_SEVERE, 1,
: 1137      1736      3      .DESC_ADDR, .IO_STS, .IO_STV);
: 1138      1737      3
: 1139      1738      3      STR$FREE1_DX (.DESC_ADDR);
: 1140      1739      3      %FI
: 1141      1740      3
: 1142      1741      3      RETURN (.IO_STS);
: 1143      1742      2      END;
: 1144      1743      2
: 1145      1744      2      [INRANGE, OUTRANGE] :
: 1146      1745      2      ASSERT (0);
: 1147      1746      2      TES;
: 1148      1747      2
: 1149      1748      2      ASSERT (0);
: 1150      1749      2      RETURN (0);
: 1151      1750      1      END;
```

! of routine EDT\$FILEIO

.TITLE EDT\$FILEIO FILEIO - Central file I/O module  
.IDENT \V04-000\

.PSECT \_EDT\$DATA,NOEXE, PIC,2

```
0000 00000 CMD_DESC:
02 0E 00002 .WORD 0
00000000 00004 .BYTE 14, 2
0000 00008 JOU_DESC:
02 0E 0000A .WORD 0
00000000 0000C .BYTE 14, 2
0000 00010 INP_DESC:
02 0E 00012 .WORD 0
00000000 00014 .BYTE 14, 2
0000 00018 ALT_DESC:
02 0E 0001A .WORD 0
00000000 0001C .BYTE 14, 2
0000 00020 OUT_DESC:
02 0E 00022 .WORD 0
00000000 00024 .BYTE 14, 2
0000 00028 INP_NAME:
00028 INP_NAME .BLKB 256
00128 INP_NAME_LEN:
0012C OUT_IFI: .BLKB 4
00130 JOU_IFI: .BLKB 4
00134 INCC_IFI:
```



EDT\$FILEIO  
V04-000

FILEIO - Central file I/O module  
EDT\$FILEIO - Central EDT file I/O routine

F 13  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILEIO.BLI;1 Page 23  
(3)

```

                                .BLKB 4
00138 INP_IFI:.BLKB 4
0013C CMD_IFI:.BLKB 4
00140 DISK_FI:.BLKB 4
                                00000000 00144 FLUSH_COUNTER:
                                .LONG 0
00148 INCL_VFC:
                                .BLKB 4
0014C INPUT_VFC:
                                .BLKB 4
                                .PSECT _EDT$CODE,NOWRT, SHR, PIC,2
50 4D 54 2E 00000 P.AAA: .ASCII \.TMP\
                                EDT$K_OPEN_INPUT== 1
                                EDT$K_OPEN_OUTPUT_SEQ==
                                2
                                EDT$K_OPEN_OUTPUT_NOSEQ==
                                3
                                EDT$K_OPEN_IN_OUT== 4
                                EDT$K_GET== 5
                                EDT$K_PUT== 6
                                EDT$K_CLOSE_DEL== 7
                                EDT$K_CLOSE== 8
                                EDT$K_COMMAND_FILE==1
                                EDT$K_INPUT_FILE== 2
                                EDT$K_INCLUDE_FILE==3
                                EDT$K_JOURNAL_FILE==4
                                EDT$K_OUTPUT_FILE== 5
                                EDT$K_WRITE_FILE== 6
                                .EXTRN EDT$$PAR_FNAME, EDT$$CNV_UPC
                                .EXTRN EDT$$REN_FI, EDT$$FLUSH_OBUF
                                .EXTRN EDT$$OPN_IFIDEF
                                .EXTRN EDT$$OPN_OFIDEF
                                .EXTRN EDT$$CLS_FI, EDT$$RD_IFI
                                .EXTRN EDT$$WR_OFI, STR$FREE1_DX
                                .EXTRN EDT$$OPN_INOUT, STR$COPY_DX
                                .EXTRN STR$COPY_R, EDT$$Z_SYS_PRIAB
                                .EXTRN EDT$$Z_SYS_JOURAB
                                .EXTRN EDT$$Z_SYS_CMDRAB
                                .EXTRN EDT$$Z_SYS_ALTRAB
                                .EXTRN EDT$ _INPFICOPN, EDT$ _FILNAM
                                .EXTRN EDT$ _INTERERR, EDT$ _COMFILNEX
                                .EXTRN EDT$ _COMFILNOP, EDT$ _NOJNLFIL
                                .EXTRN EDT$ _INPFILNEX, EDT$ _OUTFILCRE
                                .EXTRN EDT$ _NONSTDFIL, EDT$$INTER_ERR
                                OFFC 00000
                                .ENTRY EDT$FILEIO, Save R2,R3,R4,R5,R6,R7,R8,R9,- ; 0818
                                R10,R11
                                MOVAB EDT$$Z_SYS_ALTRAB, R11
                                MOVAB LIB$STOP, R10
                                MOVAB ALT_DESC, R9
                                SUBL2 #20, SP
                                MOVL FILE_REC, R6 ; 0995
                                MOVL FILE_RHB, R2 ; 0996
                                CASEL @FILECODE, #1, #7 ; 1002
```

07

025F 03D8	0125 04ED	0125 0346	0012 02B6	00027 1\$: 0002F	.WORD	2\$-1\$,- 15\$-1\$,- 15\$-1\$,- 28\$-1\$,- 33\$-1\$,- 41\$-1\$,- 68\$-1\$,- 51\$-1\$	
			0E 11 00037 08 AE D4 00039 2\$: 08 BC CF 0003C 0009 00041 3\$:	BRB CLRL CASEL .WORD	4\$ NONSTD @FILESTRM, #1, #2 5\$-3\$,- 8\$-3\$,- 12\$-3\$		
	02 00C2	01 005C	0289 31 00047 4\$: 08 AE 9F 0004A 5\$: 08 AE 9F 0004D 14 AE 9F 00050 1C AE 9F 00053 7E D4 00056 04 A2 DD 00058 56 DD 0005E 00000000G 00 9F 00060 00000000G 00 9F 00066 0124 C9 50 D0 0006D 13 12 00072 0C AE DD 00074 14 AE DD 00077 56 DD 0007A 01 DD 0007C 0085109C 8F DD 0007E 6A 05 FB 00084 08 AE E9 00087 6\$: 10 AE 00000000G 8F D0 0008B 04 A6 DD 00093 7\$: 56 DD 00096 E8 A9 9F 00098 63 11 0009B 08 AE 9F 0009D 8\$: 0134 C9 9F 000A0 14 AE 9F 000A4 1C AE 9F 000A7 7E D4 000AA 04 A2 DD 000AC 56 DD 000B2 00000000G 00 9F 000B4 00000000G 00 9F 000BA 0120 C9 50 D0 000C1 0110 C9 66 3C 000C6 10 A9 04 B6 0110 C9 28 000CB 0120 C9 D5 000D3 13 12 000D7 0C AE DD 000D9 14 AE DD 000DC 56 DD 000DF	BRW PUSHAB PUSHAB PUSHAB PUSHAB CLRL MOVZWL PUSHL PUSHL PUSHAB CALLS MOVL BNEQ PUSHL PUSHL PUSHL PUSHL PUSHL PUSHL CALLS BLBC MOVL PUSHL PUSHL PUSHAB BRB PUSHAB PUSHAB PUSHAB CLRL MOVZWL PUSHL PUSHL PUSHAB CALLS MOVL MOVZWL MOVC3 TSTL BNEQ PUSHL PUSHL PUSHL	32\$ NONSTD VFC IO_STV IO_STS -(SP) (R2), -(SP) 4(R2) R6 EDT\$\$Z SYS CMDRAB #9, EDT\$\$OPN_IFIDEF R0, CMD_IFI 6\$ IO_STV IO_STS R6 #1 #8720540 #5, LIB\$STOP NONSTD, 7\$ #EDT\$NONSTDFIL, IO_STS 4(R6) R6 CMD_DESC 11\$ NONSTD INPUT_VFC IO_STV IO_STS -(SP) (R2), -(SP) 4(R2) R6 EDT\$\$Z SYS PRIRAB #9, EDT\$\$OPN_IFIDEF R0, INP_IFI (R6), INP_NAME_LEN INP_NAME_LEN, 34(R6), INP_NAME INP_IFI 9\$ IO_STV IO_STS R6	1745 1019 1021  1140 1029  1030 1029  1035 1041 1040  1047 1052  1067  1068 1067  1072 1073 1078 1081 1080	



		01	DD	000E1	PUSHL	#1		
		8F	DD	000E3	PUSHL	#8720540		
	6A	05	FB	000E9	CALLS	#5, LIB\$STOP		
	08	08	AE	E9 000EC	9\$: BLBC	NONSTD, 10\$		1087
10	AE	00000000G	8F	D0 000F0	10\$: MOVL	#EDT\$_NONSTDFIL, IO_STS		
		04	A6	DD 000F8	PUSHL	4(R6)		1098
			56	DD 000FB	PUSHL	R6		
		F8	A9	9F 000FD	PUSHAB	INP_DESC		
		01C6	31	00100	11\$: BRW	31\$		
		08	AE	9F 00103	12\$: PUSHAB	NONSTD		1108
	0130	C9	9F	00106	PUSHAB	INCL_VFC		
	14	AE	9F	0010A	PUSHAB	IO_STV		
	1C	AE	9F	0010D	PUSHAB	IO_STS		
		01	DD	00110	PUSHL	#1		
	7E		62	3C 00112	MOVZWL	(R2), -(SP)		1109
		04	A2	DD 00115	PUSHL	4(R2)		1108
			56	DD 00118	PUSHL	R6		
			5B	DD 0011A	PUSHL	R11		
00000000G	00	09	FB	0011C	CALLS	#9, EDT\$OPN_IFIDEF		
011C	C9	50	D0	00123	MOVL	R0, INCL_IFI		
		13	12	00128	BNEQ	13\$		1111
		0C	AE	DD 0012A	PUSHL	IO_STV		1117
		14	AE	DD 0C12D	PUSHL	IO_STS		
			56	DD 00130	PUSHL	R6		1116
		01	DD	00132	PUSHL	#1		
		0085109C	8F	DD 00134	PUSHL	#8720540		
	6A	05	FB	0013A	CALLS	#5, LIB\$STOP		
	08	08	AE	E9 0013D	13\$: BLBC	NONSTD, 14\$		1123
10	AE	00000000G	8F	D0 00141	MOVL	#EDT\$_NONSTDFIL, IO_STS		
		00F5	31	00149	14\$: BRW	26\$		1128
	02	04	BC	D1 0014C	15\$: CMPL	@FILECODE, #2		1159
			05	12 00150	BNEQ	16\$		
	58		01	D0 00152	MOVL	#1, SEQ		1161
			02	11 00155	BRB	17\$		
			58	D4 00157	16\$: CLRL	SEQ		1163
02	04	08	BC	CF 00159	17\$: CASEL	@FILESTRM, #4, #2		1165
0009	0009	00EC		0015E	18\$: .WORD	27\$-18\$,-		
						19\$-18\$,-		
						19\$-18\$		
		016C	31	00164	BRW	32\$		1352
	05	08	BC	D1 00167	19\$: CMPL	@FILESTRM, #5		1176
			0A	12 00168	BNEQ	20\$		
			57	D4 0016D	CLRL	ATT		1179
	55		01	D0 0016F	MOVL	#1, RELAT		1180
	53		01	D0 00172	MOVL	#1, FORCE_MAXV		1181
			07	11 00175	BRB	21\$		1176
	57		01	D0 00177	20\$: MOVL	#1, ATT		1185
			55	D4 0017A	CLRL	RELAT		1186
			53	D4 0017C	CLRL	FORCE_MAXV		1187
			62	B5 0017E	21\$: TSTW	(R2)		1193
			07	13 00180	BEQL	22\$		
00000000G	00	00	FB	00182	CALLS	#0, EDT\$INTER_ERR		
		0128	C9	D4 00189	22\$: CLRL	DISK_FI		1194
	01		55	D1 0018D	CMPL	RELAT, #1		1204
			14	12 00190	BNEQ	23\$		
			66	B5 00192	TSTW	(R6)		
			10	12 00194	BNEQ	23\$		

		10	A9	9F	00196	PUSHAB	INP_NAME	:	1207
		0110	C9	9F	00199	PUSHAB	INP_NAME_LEN	:	
			56	DD	0019D	PUSHL	R6	:	
00000000G	00		03	FB	0019F	CALLS	#3, STR\$COPY_R	:	
		0C	AE	9F	001A6	23\$: PUSHAB	IO_STV	:	1214
		14	AE	9F	001A9	PUSHAB	IO_STS	:	
		0128	C9	9F	001AC	PUSHAB	DISK_FI	:	
			55	DD	001B0	PUSHL	RELAT	:	
			56	DD	001B2	PUSHL	R6	:	
			5B	DD	001B4	PUSHL	R11	:	
00000000G	00		06	FB	001B6	CALLS	#6, EDT\$SPAR_FNAME	:	
	54		50	DD	001BD	MOVL	R0, STATUS	:	
	13		54	E8	001C0	BLBS	STATUS, 24\$	:	1216
		0C	AE	DD	001C3	PUSHL	IO_STV	:	1219
		14	AE	DD	001C6	PUSHL	IO_STS	:	
			56	DD	001C9	PUSHL	R6	:	1218
			01	DD	001CB	PUSHL	#1	:	
	008510A4		8F	DD	001CD	PUSHL	#8720548	:	
	6A		05	FB	001D3	CALLS	#5, LIB\$STOP	:	
		08	A9	B4	001D6	24\$: CLRW	OUT_DESC	:	1221
		0C	A9	D4	001D9	CLRL	OUT_DESC+4	:	1222
	26	0128	C9	E9	001DC	BLBC	DISK_FI, 25\$	:	1228
		04	A6	DD	001E1	PUSHL	4(R6)	:	1231
			56	DD	001E4	PUSHL	R6	:	
		08	A9	9F	001E6	PUSHAB	OUT_DESC	:	
00000000G	00		03	FB	001E9	CALLS	#3, STR\$COPY_R	:	
		FE08	CF	9F	001F0	PUSHAB	P.AAA	:	1232
	04	AE						:	
		04	AE	9F	001F8	MOVL	#4, 4(SP)	:	
			56	DD	001FB	PUSHAB	4(SP)	:	
00000000G	00		03	FB	001FD	PUSHL	R6	:	
	53		01	DD	00204	CALLS	#3, STR\$COPY_R	:	1233
		0C	AE	9F	00207	25\$: MOVL	#1, FORCE_MAXV	:	1240
		14	AE	9F	0020A	PUSHAB	IO_STV	:	
			53	DD	0020D	PUSHAB	IO_STS	:	
		00A0	8F	BB	0020F	PUSHL	FORCE_MAXV	:	1241
			58	DD	00213	PUSHR	#*M<R5,R7>	:	
	7E	08	A9	3C	00215	PUSHL	SEQ	:	
		0C	A9	DD	00219	MOVZWL	OUT_DESC, -(SP)	:	
			56	DD	0021C	PUSHL	OUT_DESC+4	:	1240
			5B	DD	0021E	PUSHL	R6	:	
00000000G	00		0A	FB	00220	PUSHL	R11	:	
	0114	C9	50	DD	00227	CALLS	#10, EDT\$OPN_OF IDEF	:	
			13	12	0022C	MOVL	R0, OUT_IFI	:	
		0C	AE	DD	0022E	BNEQ	26\$	:	1246
		14	AE	DD	00231	PUSHL	IO_STV	:	1249
			56	DD	00234	PUSHL	IO_STS	:	
			01	DD	00236	PUSHL	R6	:	1248
	008510A4		8F	DD	00238	PUSHL	#1	:	
	6A		05	FB	0023E	PUSHL	#8720548	:	
		04	A6	DD	00241	CALLS	#5, LIB\$STOP	:	
			56	DD	00244	PUSHL	4(R6)	:	1254
			59	DD	00246	PUSHL	R6	:	
			7F	11	00248	PUSHL	R9	:	
		0C	AE	9F	0024A	27\$: BRB	31\$	:	
		14	AE	9F	0024D	PUSHAB	IO_STV	:	1330
			01	DD	00250	PUSHAB	IO_STS	:	
						PUSHL	#1	:	



EDT\$FILEIO  
V04-000

FILEIO - Central file I/O module  
EDT\$FILEIO - Central EDT file I/O routine

J 13  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILEIO.BLI;1

Page 27  
(3)

ED  
VC

			02	DD	00252	PUSHL	#2	:		
			01	DD	00254	PUSHL	#1	:		
			7E	D4	00256	CLRL	-(SP)	:		
	7E		62	3C	00258	MOVZWL	(R2), -(SP)	:	1331	
		04	A2	DD	0025B	PUSHL	4(R2)	:	1330	
			56	DD	0025E	PUSHL	R6	:		
	00000000G	00	00	9F	00260	PUSHAB	EDT\$\$Z SYS JOURAB	:		
	0118	C9	0A	FB	00266	CALLS	#10, EDT\$\$OPN_OFIDEF	:		
			50	DO	0026D	MOVL	R0, JOU_IFI	:		
			4D	12	00272	BNEQ	30\$	:	1333	
		0C	AE	DD	00274	PUSHL	IO_STV	:	1336	
		14	AE	DD	00277	PUSHL	IO_STS	:		
			56	DD	0027A	PUSHL	R6	:	1335	
			01	DD	0027C	PUSHL	#1	:		
		008510A4	8F	DD	0027E	PUSHL	#8720548	:		
			38	11	00284	BRB	29\$	:		
	04	08	BC	D1	00286	CMPL	@FILESTRM, #4	:	1366	
			47	12	0028A	BNEQ	32\$	:		
		0C	AE	9F	0028C	PUSHAB	IO_STV	:	1372	
		14	AE	9F	0028F	PUSHAB	IO_STS	:		
	7E		62	3C	00292	MOVZWL	(R2), -(SP)	:	1373	
		04	A2	DD	00295	PUSHL	4(R2)	:	1372	
			56	DD	00298	PUSHL	R6	:		
		00000000G	00	9F	0029A	PUSHAB	EDT\$\$Z SYS JOURAB	:		
	00000000G	00	06	FB	002A0	CALLS	#6, EDT\$\$OPN_INOUT	:		
	0118	C9	50	DO	002A7	MOVL	R0, JOU_IFI	:		
			13	12	002AC	BNEQ	30\$	:	1375	
		0C	AE	DD	002AE	PUSHL	IO_STV	:	1378	
		14	AE	DD	002B1	PUSHL	IO_STS	:		
			56	DD	002B4	PUSHL	R6	:	1377	
			01	DD	002B6	PUSHL	#1	:		
		0085109C	8F	DD	002B8	PUSHL	#8720540	:		
	6A		05	FB	002BE	CALLS	#5, LIB\$STOP	:		
		04	A6	DD	002C1	PUSHL	4(R6)	:	1380	
			56	DD	002C4	PUSHL	R6	:		
		F0	A9	9F	002C6	PUSHAB	JOU_DESC	:		
	00000000G	00	03	FB	002C9	CALLS	#3, STR\$COPY_R	:		
			02AA	31	002D0	BRW	76\$	:	1387	
	00000000G	00	00	FB	002D3	CALLS	#0, EDT\$\$INTER_ERR	:	1390	
			02A5	31	002DA	BRW	77\$	:	1002	
	03	01	08	BC	CF	CASEL	@FILESTRM, #1, #3	:	1401	
0042	0034	0021	0011		002E2	.WORD	35\$-34\$,-	:		
							36\$-34\$,-	:		
							37\$-34\$,-	:		
							38\$-34\$	:		
	00000000G	00	00	FB	002EA	CALLS	#0, EDT\$\$INTER_ERR	:	1433	
			3F	11	002F1	BRB	39\$	:	1401	
		53	E8	A9	9E	002F3	MOVAB	CMD_DESC, DESC_ADDR	:	1406
	50	00000000G	00	9E	002F7	MOVAB	EDT\$\$Z_SYS_CMDRAB, RAB	:	1407	
		04	AE	D4	002FE	CLRL	VFC	:	1408	
			2F	11	00301	BRB	39\$	:	1401	
		53	F8	A9	9E	00303	MOVAB	INP_DESC, DESC_ADDR	:	1413
	04	AE	0134	C9	DO	00307	MOVL	INPOT_VFC, VFC	:	1414
		50	00000000G	00	9E	0030D	MOVAB	EDT\$\$Z_SYS_PRIAB, RAB	:	1415
			1C	11	00314	BRB	39\$	:	1401	
	04	AE	0130	C9	DO	00316	MOVL	INCL_VFC, VFC	:	1420
		53		69	9E	0031C	MOVAB	ALT_DESC, DESC_ADDR	:	1421

50		6B	9E	0031F	MOVAB	EDT\$Z_SYS_ALTRAB, RAB	1422	
		0E	11	00322	BRB	39\$	1401	
	04	AE	D4	00324	CLRL	VFC	1427	
53	F0	A9	9E	00327	MOVAB	JOU DESC, DESC ADDR	1428	
50	00000000G	00	9E	0032B	MOVAB	EDT\$Z_SYS_JOURAB, RAB	1429	
	04	AE	DD	00332	PUSHL	VFC	1438	
	10	AE	9F	00335	PUSHAB	IO_STV		
	18	AE	9F	00338	PUSHAB	IO_STS		
		52	DD	0033B	PUSHL	R2		
	0041	8F	BB	0033D	PUSHR	#^M<R0,R6>		
00000000G	00	06	FB	00341	CALLS	#6, EDT\$SRD_IF1		
	54	50	DD	00348	MOVL	R0, STATUS		
	72	54	E8	0034B	BLBS	STATUS, 47\$	1440	
0001827A	8F	10	AE	D1	0034E	CMPL	IO_STS, #98938	1443
		03	12	00356	BNEQ	40\$		
		0222	31	00358	BRW	76\$		
	0C	AE	DD	0035B	PUSHL	IO_STV	1448	
	14	AE	DD	0035E	PUSHL	IO_STS	1447	
		53	DD	00361	PUSHL	DESC_ADDR		
		01	DD	00363	PUSHL	#1		
	008510B4	8F	DD	00365	PUSHL	#8720564		
		50	11	0036B	BRB	46\$		
02	04	08	BC	CF	CASEL	@FILESTRM, #4, #2	1473	
000F	000F	0017		00372	.WORD	44\$-42\$,-		
						43\$-42\$,-		
						43\$-42\$		
00000000G	00	00	FB	00378	CALLS	#0, EDT\$INTER_ERR	1489	
		13	11	0037F	BRB	45\$	1473	
	55	69	9E	00381	MOVAB	ALT DESC, DESC ADDR	1478	
	53	6B	9E	00384	MOVAB	EDT\$Z_SYS_ALTRAB, RAB	1479	
		0B	11	00387	BRB	45\$	1473	
	55	F0	A9	9E	00389	MOVAB	JOU DESC, DESC ADDR	1484
	53	00000000G	00	9E	0038D	MOVAB	EDT\$Z_SYS_JOURAB, RAB	1485
		0C	AE	9F	00394	PUSHAB	IO_STV	1494
		14	AE	9F	00397	PUSHAB	IO_STS	
		52	DD	0039A	PUSHL	R2		
	0048	8F	BB	0039C	PUSHR	#^M<R3,R6>		
00000000G	00	05	FB	003A0	CALLS	#5, EDT\$SWR_OF1		
	54	50	DD	003A7	MOVL	R0, STATUS		
	15	54	E8	003AA	BLBS	STATUS, 48\$	1500	
		0C	AE	DD	003AD	PUSHL	IO_STV	1506
		14	AE	DD	003B0	PUSHL	IO_STS	1505
		55	DD	003B3	PUSHL	DESC_ADDR		
		01	DD	003B5	PUSHL	#1		
	008510D4	8F	DD	003B7	PUSHL	#8720596		
6A		05	FB	003BD	CALLS	#5, LIB\$STOP		
		39	11	003C0	BRB	50\$		
04	08	BC	D1	003C2	CMPL	@FILESTRM, #4	1511	
		33	12	003C6	BNEQ	50\$		
	012C	C9	D6	003C8	INCL	FLUSH_COUNTER	1514	
	05	012C	C9	D1	CMPL	FLUSH_COUNTER, #5	1516	
		28	12	003D1	BNEQ	50\$		
		0C	AE	9F	PUSHAB	IO_STV	1522	
		53	DD	003D6	PUSHL	RAB		
00000000G	00	02	FB	003D8	CALLS	#2, EDT\$FLUSH_OBUF		
	54	50	DD	003DF	MOVL	R0, STATUS		
	12	54	E8	003E2	BLBS	STATUS, 49\$	1524	



EDT\$FILEIO  
V04-000

FILEIO - Central file I/O module  
EDT\$FILEIO - Central EDT file I/O routine

L 13  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILEIO.BLI;1

Page 29  
(3)

		0C	AE	DD	003E5	PUSHL	IO_STV	1527		
			54	DD	003E8	PUSHL	STATUS			
			55	DD	003EA	PUSHL	DESC_ADDR	1526		
			01	DD	003EC	PUSHL	#1			
	008510D4	8F	DD	003EE	PUSHL	#8720596				
6A		05	FB	003F4	CALLS	#5, LIB\$STOP				
	012C	C9	D4	003F7	CLRL	FLUSH COUNTER		1533		
50		54	D0	003FB	MOVL	STATUS, R0		1538		
			04	003FE	RET					
		08	BC	CF	CASEL	@FILESTRM, #1, #5		1548		
00D6	05	01								
	0054	0034	0015	00404	52\$:	.WORD	53\$-52\$,-			
		0075	0075	0040C			54\$-52\$,-			
							56\$-52\$,-			
							64\$-52\$,-			
							58\$-52\$,-			
							58\$-52\$			
	00000000G	00	00	FB	00410	CALLS	#0, EDT\$\$INTER_ERR	1671		
			5E	11	00417	BRB	57\$	1548		
		52	E8	A9	9E	00419	53\$:	MOVAB	CMD_DESC, DESC_ADDR	1556
		53	1050	8F	3C	0041D	MOVZWL	#4176, ERROR	1557	
			0C	AE	9F	00422	PUSHAB	IO_STV	1558	
			14	AE	9F	00425	PUSHAB	IO_STS		
				52	DD	00428	PUSHL	DESC_ADDR		
				7E	D4	0042A	CLRL	-(SP)		
	00000000G	00	00	9F	0042C	PUSHAB	EDT\$\$Z_SYS_CMDRAB			
		0124	C9	DD	00432	PUSHL	CMD_IFI			
			1D	11	00436	BRB	55\$			
		52	F8	A9	9E	00438	54\$:	MOVAB	INP_DESC, DESC_ADDR	1570
		53	1050	8F	3C	0043C	MOVZWL	#4176, ERROR	1571	
			0C	AE	9F	00441	PUSHAB	IO_STV	1572	
			14	AE	9F	00444	PUSHAB	IO_STS		
				52	DD	00447	PUSHL	DESC_ADDR		
				7E	D4	00449	CLRL	-(SP)		
	00000000G	00	00	9F	0044B	PUSHAB	EDT\$\$Z_SYS_PRIAB			
		0120	C9	DD	00451	PUSHL	INP_IFI			
			009F	31	00455	BRW	65\$			
		52	69	9E	00458	55\$:	MOVAB	ALT_DESC, DESC_ADDR	1584	
						56\$:	PUSHAB	IO_STV	1585	
			0C	AE	9F	0045B	PUSHAB	IO_STS		
			14	AE	9F	0045E	PUSHAB	IO_STS		
				52	DD	00461	PUSHL	DESC_ADDR		
				7E	D4	00463	CLRL	-(SP)		
				5B	DD	00465	PUSHL	R11		
		011C	C9	DD	00467	PUSHL	INCL_IFI			
	00000000G	00	06	FB	0046B	CALLS	#6, EDT\$\$CLS_FI			
		53	1050	8F	3C	00472	MOVZWL	#4176, ERROR	1586	
				5F	11	00477	57\$:	BRB	63\$	1548
		05	08	BC	D1	00479	58\$:	CMPL	@FILESTRM, #5	1599
				05	12	0047D	BNEQ	59\$		
		54		01	D0	0047F	MOVL	#1, FORCE_MAXV		
				02	11	00482	BRB	60\$		
				54	D4	00484	59\$:	CLRL	FORCE_MAXV	
		52		69	9E	00486	60\$:	MOVAB	ALT_DESC, DESC_ADDR	1603
		53	1058	8F	3C	00489	MOVZWL	#4184, ERROR	1604	
			0C	AE	9F	0048E	PUSHAB	IO_STV	1605	
			14	AE	9F	00491	PUSHAB	IO_STS		
				52	DD	00494	PUSHL	DESC_ADDR		
				7E	D4	00496	CLRL	-(SP)		

00000000G	00	0114	5B	DD	00498	PUSHL	R11	:	
	59		C9	DD	0049A	PUSHL	OUT_IFI	:	
	1C		06	FB	0049E	CALLS	#6, EDT\$\$CLS_FI	:	1610
		10	AE	E9	004A5	BLBC	IO_STS, 67\$	:	1613
		0128	C9	E9	004A9	BLBC	DISK_FI, 61\$	:	1616
		0C	AE	9F	004AE	PUSHAB	IO_STV	:	
		14	AE	9F	004B1	PUSHAB	IO_STS	:	
			54	DD	004B4	PUSHL	FORCE_MAXV	:	
		08	A9	9F	004B6	PUSHAB	OUT_DESC	:	
			59	DD	004B9	PUSHL	R9	:	
00000000G	00		05	FB	004BB	CALLS	#5, EDT\$\$REN_FI	:	
		0C	A9	DD	004C2	PUSHL	OUT_DESC+4	:	1617
		08	A9	9F	004C5	PUSHAB	OUT_DESC	:	
			05	11	004C8	BRB	62\$	:	
		04	A9	DD	004CA	PUSHL	ALT_DESC+4	:	1620
			59	DD	004CD	PUSHL	R9	:	
			56	DD	004CF	PUSHL	R6	:	
00000000G	00		03	FB	004D1	CALLS	#3, STR\$COPY_R	:	
			24	11	004DB	BRB	66\$	:	1613
	52		A9	9E	004DA	MOVAB	JOU_DESC, DESC_ADDR	:	1661
	53	1058	8F	3C	004DE	MOVZWL	#4184, ERROR	:	1662
		0C	AE	9F	004E3	PUSHAB	IO_STV	:	1663
		14	AE	9F	004E6	PUSHAB	IO_STS	:	
			52	DD	004E9	PUSHL	DESC_ADDR	:	
			7E	D4	004EB	CLRL	-(SP)	:	
		00000000G	00	9F	004ED	PUSHAB	EDT\$\$Z_SYS_JOURAB	:	
		0118	C9	DD	004F3	PUSHL	JOU_IFI	:	
00000000G	00		06	FB	004F7	CALLS	#6, EDT\$\$CLS_FI	:	
	72	10	AE	E8	004FE	BLBS	IO_STS, 75\$	:	1680
		0C	AE	DD	00502	PUSHL	IO_STV	:	1683
		14	AE	DD	00505	PUSHL	IO_STS	:	
			52	DD	00508	PUSHL	DESC_ADDR	:	1682
			01	DD	0050A	PUSHL	#1	:	
		00850004	E3	9F	0050C	PUSHAB	8716292(ERROR)	:	
			5D	11	00512	BRB	74\$	:	
02	04	08	BC	CF	00514	CASEL	@FILESTRM, #4, #2	:	1697
000F	000F	0024			00519	.WORD	71\$-69\$,-	:	
							70\$-69\$,-	:	
							70\$-69\$	:	
00000000G	00		00	FB	0051F	CALLS	#0, EDT\$\$INTER_ERR	:	1727
			35	11	00526	BRB	73\$	:	1697
	52		69	9E	00528	MOVAB	ALT_DESC, DESC_ADDR	:	1705
		0C	AE	9F	0052B	PUSHAB	IO_STV	:	1706
		14	AE	9F	0052E	PUSHAB	IO_STS	:	
			59	DD	00531	PUSHL	R9	:	
			01	DD	00533	PUSHL	#1	:	
			5B	DD	00535	PUSHL	R11	:	
		0114	C9	DD	00537	PUSHL	OUT_IFI	:	
			19	11	0053B	BRB	72\$	:	
	52		A9	9E	0053D	MOVAB	JOU_DESC, DESC_ADDR	:	1718
		0C	AE	9F	00541	PUSHAB	IO_STV	:	1719
		14	AE	9F	00544	PUSHAB	IO_STS	:	
		FO	A9	9F	00547	PUSHAB	JOU_DESC	:	
			02	DD	0054A	PUSHL	#2	:	
		00000000G	00	9F	0054C	PUSHAB	EDT\$\$Z_SYS_JOURAB	:	
		0118	C9	DD	00552	PUSHL	JOU_IFI	:	
00000000G	00		06	FB	00556	CALLS	#6, EDT\$\$CLS_FI	:	



EDT\$FILE10  
V04-000

FILE10 - Central file I/O module  
EDT\$FILE10 - Central EDT file I/O routine

N 13  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILE10.BLI;1

Page 31  
(3)

13	10	AE	E8	0055D	73\$:	BLBS	IO_STS, 75\$	:	1733	
	0C	AE	DD	00561		PUSHL	IO_STV	:	1736	
	14	AE	DD	00564		PUSHL	IO_STS	:		
		52	DD	00567		PUSHL	DESC_ADDR	:		
		01	DD	00569		PUSHL	#1	:	1735	
	0085105C	8F	DD	0056B		PUSHL	#8720476	:		
6A		05	FB	00571	74\$:	CALLS	#5, LIB\$STOP	:		
		52	DD	00574	75\$:	PUSHL	DESC_ADDR	:	1738	
00000000G	00	01	FB	00576		CALLS	#1, STR\$FREE1_DX	:		
	50	10	AE	D0	0057D	76\$:	MOVL	IO_STS, R0	:	1741
				04	00581	RET		:		
00000000G	00	00	FB	00582	77\$:	CALLS	#0, EDT\$\$INTER_ERR	:	1748	
		50	D4	00589		CLRL	R0	:	1749	
			04	0058B		RET		:	1750	

; Routine Size: 1420 bytes, Routine Base: \_EDT\$CODE + 0004

; 1152 1751 1  
; 1153 1752 1 !<BLF/PAGE>

EDT\$FILEIO  
V04-000

FILEIO - Central file I/O module  
EDT\$FILEIO - Central EDT file I/O routine

B 14  
16-Sep-1984 00:21:05  
14-Sep-1984 12:23:06

VAX-11 Bliss-32 V4.0-742  
DISK\$VMSMASTER:[EDT.SRC]FILEIO.BLI;1 Page 32  
(4)

: 1155 1753 1 END  
: 1156 1754 1  
: 1157 1755 0 ELUDOM

! of module EDT\$FILEIO

.EXTRN LIB\$STOP

# PSECT SUMMARY

Name	Bytes	Attributes
EDT\$DATA	336	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, PIC, ALIGN(2)
EDT\$CODE	1424	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)
ABS	0	NOVEC, NOWRT, NORD, NOEXE, NOSHR, LCL, ABS, CON, NOPIC, ALIGN(0)

# Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[EDT.SRC]EDT.L32;1	377	3	0	40	00:00.2
\$255\$DUA28:[EDT.SRC]PSECTS.L32;1	2	1	50	7	00:00.1
\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	14	0	581	00:04.1

# COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:FILEIO/OBJ=OBJ\$:FILEIO MSRC\$:FILEIO.BLI/UPDATE=(ENH\$:FILEIO)

: Size: 1420 code + 340 data bytes

: Run Time: 01:06.6

: Elapsed Time: 01:23.8

: Lines/CPU Min: 1581

: Lexemes/CPU-Min: 7978

: Memory Used: 357 pages

: Compilation Complete



0133 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

EXTEND  
LIS

FDEC  
LIS

FILL  
LIS

FINDPARA  
LIS

FCRLF  
LIS

EDT  
LIS

EXEC  
LIS

EXECNOO  
LIS

FILEIO  
LIS

EDTVECTOR  
LIS

FINDKEY  
LIS

FCOLINC  
LIS

FINAL  
LIS

FINDHDLR  
LIS

DEFKEY  
LIS

ERRMSG  
LIS

FCHAR  
LIS